BIREME / OPS / OMS

Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud

Lenguaje de Formato CISIS

4.x

São Paulo - 2002 - 2006

Ficha Catalográfica

BIREME / OPS / OMS (Brasil)

Lenguaje de Formato CISIS. / BIREME (org.). São Paulo : BIREME / OPS / OMS, 2002 - 2006.

44 p.

1. Manual del usuario. 2. Acceso a la información. 3. Sistemas de información. 4. Gerenciamiento de información. 5. Salud Pública. 6. Servicios de salud . I. BIREME II. Título

Advertencia - La mención a las compañías y/o instituciones específicas o a ciertos productos no implica que estos sean apoyados o recomendados por BIREME / OPS / OMS, y no significa que haya preferencia en relación a otros de naturaleza similar, citados o no.

BIREME / OPS / OMS

Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud

Rua Botucatu, 862 - V. Clementino

Este documento fue producido con la Metodología para la Normalización de Documentos (NorDoc) desarrollada por BIREME.

Tabla de contenido

Abreviaturas utilizadas

- **ANSI**. American National Standards Institute [Instituto Nacional Americano de Normas].
- **ASCII**. American Standard Code for Information Interchange [Código Americano Normalizado para el Intercambio de Información].
- **BIREME**. Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud.
- **BVS**. Biblioteca Virtual en Salud.
- **CDS**. Computerized Documentation System [Sistema de Documentación Computarizada].
- **FST**. Field Selection Table [Tabla de Selección de Campo].
- **FTP**. File Transfer Protocol [Protocolo de transferencia de archivos].
- **IFP**. Inverted File Pointer [Puntero de archivo invertido].

- **ISIS**. Integrated Set of Information Systems [Conjunto integrado de sistemas de información].
- **ISO**. International Organization for Standardization [Organización Internacional para la Normalización].
- **LILACS**. Literatura Latinoamericana y del Caribe en Ciencias de la Salud.
- **OMS**. Organización Mundial de la Salud.
- **OPS**. Organización Panamericana de la Salud.
- **UNESCO**. United Nations Educational, Scientific and Cultural Organization [Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura].

Como usar este manual

Este manual está organizado en orden alfabética de comando o función. También están listados en orden alfabético los nombres atribuidos a cada comando o función para ayudar el usuario con la localización de los temas deseados.

Cada comando/función tiene toda información acerca de su utilización descrita en formato de tabla, conteniendo sintaxis y ejemplos.

Complementan el documento, un glosario, un listado de abreviaturas y las citas bibliográficas para otros documentos relevantes.

Prefacio

Sobre BIREME

Año tras año, BIREME cumple su misión como centro especializado en información científica y técnica en salud para la región de América Latina y el Caribe. Establecida en Brasil en 1967, con el nombre de Biblioteca Regional de Medicina (que originó la sigla BIREME), atendió desde el inicio a la creciente demanda de literatura científica actualizada por parte de los sistemas nacionales de salud y las comunidades de investigadores, profesionales y estudiantes. Posteriormente, en 1982, pasó a llamarse Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud, para mejor expresar sus funciones, orientadas al fortalecimiento y ampliación del flujo de información científica y técnica en salud en toda la región, pero conservó su sigla.

El trabajo en red, en base a la descentralización, orientado a desarrollar capacidades locales, compartir recursos de información, desarrollar productos y servicios cooperativos, elaborar metodologías comunes, siempre fue el fundamento del trabajo de cooperación técnica de BIREME. De esa forma el centro se consolida como un modelo internacional que promueve la capacitación de los profesionales de información a nivel gerencial y técnico, para que adopten los paradigmas de información y comunicación que mejor atiendan a las necesidades locales.

Los principales fundamentos que dan origen y soporte a la existencia de BIREME son los siguientes:

- el acceso a la información científico-técnica en salud es esencial al desarrollo de la salud;
- la necesidad de desarrollar la capacidad de los países de América Latina y el Caribe de operar las fuentes de información científico-técnica en salud de forma cooperativa y eficiente;
- la necesidad de promover el uso y de responder a las demandas de información científico-técnica en salud de los gobiernos, los sistemas de salud, las instituciones de enseñanza e investigación.

BIREME, como centro especializado de la Organización Panamericana de la Salud (OPAS)/Organización Mundial de la Salud (OMS), coordina y realiza actividades de cooperación técnica en gestión de información y conocimiento científico, con el propósito de fortalecer y ampliar el flujo de información científica en salud en Brasil y en los demás países de América Latina y el Caribe, como condición esencial para el desarrollo de la salud, incluyendo planificación, gestión, promoción, investigación, educación y atención.

El convenio que fundamenta BIREME es renovado a cada cinco años por los miembros del Comité Asesor Nacional de la institución (OPAS, Ministerio de la Salud de Brasil, Ministerio de Educación y Cultura de Brasil, Secretaría de Salud del Estado de São Paulo y Universidad Federal de São Paulo – Unifesp). Esta última ofrece la infraestructura física necesaria al establecimiento de la institución.

En 2004 la institución asumió la responsabilidad de convertirse en una entidad que se basa en el conocimiento.

La Biblioteca Virtual en Salud (BVS)

Con el surgimiento y consolidación de la Internet como medio predominante de información y comunicación, el modelo de cooperación técnica de BIREME evolucionó desde 1998 hacia la construcción y desarrollo de la Biblioteca Virtual en Salud (BVS) como espacio común de convergencia del trabajo cooperativo de productores, intermediarios y usuarios de información. La BVS promueve el desarrollo de una red de fuentes de información científica y técnica con acceso

universal en la Internet. Por primera vez se abre la posibilidad real de acceso equitativo a la información en salud.

BIREME tiene a la Biblioteca Virtual como modelo para la gestión de información y conocimiento, lo que implica la cooperación y convergencia de instituciones, sistemas, redes e iniciativas de productores, intermediarios y usuarios en la operación de redes de fuentes de información locales, nacionales, regionales e internacionales, privilegiando así el acceso abierto y universal.

Actualmente, todos los países de América Latina y el Caribe (Región) participan directa o indirectamente en los productos y servicios cooperativos promovidos por la BVS, lo que involucra a más de mil instituciones en más de 30 países.

La BVS es simulada en un espacio virtual de la Internet formada por la colección o red de fuentes de información en salud de la Región. Usuarios de distintos niveles y localización pueden interactuar y navegar en el espacio de una o varias fuentes de información, independientemente de su localización física. Las fuentes de información son generadas, actualizadas, almacenadas y operadas en la Internet por productores, integradores e intermediarios, de modo descentralizado, obedeciendo a metodologías comunes para su integración a la BVS.

La BVS organiza la información en una estructura que integra e interconecta bases de datos referenciales, directorios de especialistas, eventos e instituciones, catálogo de recursos de información disponibles en la Internet, colecciones de textos completos con destaque para la colección SciELO (Scientific Electronic Online) de revistas científicas, servicios de diseminación selectiva de información, fuentes de información de apoyo a la educación y la toma de decisión, noticias, listas de discusión y apoyo a comunidades virtuales. Por lo tanto, el espacio de la BVS constituye una red dinámica de fuentes de información descentralizada a partir de la cual se puede recuperar y extraer información y conocimiento para subsidiar los procesos de decisión en el área de la salud.

La Biblioteca Virtual en Salud es visualizada como la base distribuida del conocimiento científico y técnico en salud registrado, organizado y almacenado en formato electrónico en los países de la Región, accesible de forma universal en la Internet de modo compatible con las bases internacionales.

Estructura del listado de referencia

<format specification>	**<name>**
Soporte:	Si Standard, significa que el comando/función tiene el mismo uso/resultado ambos en ISIS y CISIS. Si CISIS se especifica, significa que el comando/función sólo está disponible en CISIS. Comandos/funciones que se han mejorado en CISIS, se presenta con anotación de Standard/CISIS. (item siempre presente)
Tipo de función:	Especifica el tipo del valor de retorno de la función. Los posibles valores son: Boolean, String y Numeric. (item sólo aplicable a las funciones)
Sintaxis:	Notación formal del uso del comando/función. (item siempre presente)
Definición:	Exposición del uso del comando/función.
Componentes:	Exposición de características adicionales del comando/función.
Notas:	Aclara particularidades, restricciones y/o diferências entre ISIS y CISIS.
Ejemplos:	1 Provee uno o más ejemplos de uso del comando/función.
Vea también:	Lista comandos y funciones relacionadas.

Comandos y funciones

% " "" | | () / /* @

#	nueva línea incondicional
Soporte:	Standard
Sintaxis:	**#**
Definición:	Salta a la línea siguiente incondicionalmente.
Ejemplos:	1 **#,**("address:"v3(9,9)+|; |**#**), 2 (|Name: |v1^n,c20,|Surname:|v1^s/),**###**,
Vea también:	**/** command **%** command

%	cancela línea en blanco
Soporte:	Standard
Sintaxis:	**%**
Definición:	Cancela las líneas en blanco previas, si las ubiera.
Ejemplos:	1 |Name: |v1^n,c20,|Surname:|v1^s,###,**%**,/, 2 v10/#,v20/#,v30/#,%#,
Vea también:	**/** command **#** command

"string"	literal condicional
Soporte:	Standard

"string"	**literal condicional**
Sintaxis:	"<text>"<field selector >"<text>" "<text>"<dummy field (selector)> "<text>"<not present>
Definición:	Muestra el texto que está entre comillas dobles solo si, <field selector>, <dummy selector> o <not present>, son evaluados como TRUE (verdadero). Los literales, prefijos y sufijos, pueden colocarse junto a un <field selector> de modo que el dato contenido en el campo también se visualizará. Si se asocia a un <dummy selector>, la salida es generada sólo si el campo tiene datos. Si se asocia a <not present>, la salida es generada sólo si el campo no tiene datos.
Notas:	<text> se produce una sola vez, independientemente de la repetibilidad del campo.
Ejemplos:	1 "Author: "v1^a, 2 "this text outputs if data field 10 is present"d10, 3 "this text outputs if data field 10 is absent"n5,
Vea también:	**'string'** command **|string|** command **d** command **n** command **v** command

'string'	**literal incondicional**
Soporte:	Standard
Sintaxis:	'<texto>'
Definición:	Muestra incondicionalmente el texto contenido entre comillas simples.
Notas:	Los literales incondicionales pueden ubicarse en cualquier lugar del formato, y pueden usarse para pasar parámetros a las funciones.
Ejemplos:	1 'this text will always output', 2 'Name: ',v1/,
Vea también:	**"string"** command **|string|** command

|string|	**literal condicional repetible**
Soporte:	Standard
Sintaxis:	|<text>|<+><field selector><+>|<text>|
Definición:	Da salida al texto colocado entre las barras verticales para cada ocurrencia de un campo repetible, solamente si el selector de campo se considera TRUE (verdadero). Combinado con un campo repetible, el funcionamiento del comando puede ser ampliado mediante el uso del operador <+>. Cuando <+> está presente, el primer literal-prefijo y/o el último literal-sufijo no se visualizan.

|string| — literal condicional repetible

Notas: Los literales prefijo y sufijo pueden usarse conjuntamente, incluso con el operador <+>. Si un literal prefijo o sufijo es utilizado con <+> fuera de un grupo repetible, los contenidos del literal pueden no ser visualizados como se desea. Si el campo no es repetible, la vizualización del literal ocurre para la primera y única ocurrencia del campo de datos.

Ejemplos:

1 (|; |+v1^s,|,|v1^n*0.1|.|),

2 (v10|: |, ,v11,| - |v12),

Vea también:
"string" command
'string' command
(format**)** command
v command

(format) — grupo repetible

Soporte: Standard

Sintaxis: **(**<format>**)**

Definición: Aplica el formato contenido entre paréntesis a cada una de las ocurrencias cada campo repetible, o una sola vez, en el caso de campos no repetibles.

Notas: No está permitido el anidamiento de grupos repetibles

Ejemplos:

1 **(**|; |+v2^s/**)**,

2 **(**v1,c15,v2,c35,v3/**)**,

3 **(**if iocc<=3 then f(iocc,1,0),| - |v3/ else '-> more than 3'/, fi**)**,

Vea también:
|string| command
v command

/ — nueva línea condicional

Soporte: Standard

Sintaxis: **/**

Definición: Comienza una nueva línea siempre que no se encuentre al comienzo de una.

Notas: Varios comandos de salto de línea condicional (/,/,/,/) producen el mismo efecto que uno solo.

Ejemplos:

1 v1**/**,

2 v1**/**,v3**/**,v10**/**,mfn**/**,

3 s(v1,v3,v10)**/**,

Vea también:
command
% command

/*string*/ — comentarios

Soporte: CISIS

Sintaxis: **/*** <comment> ***/**

Definición: Encierra comentarios

Notas: Los comentarios pueden ocupar varias líneas.

Ejemplos:	1	**/*** this is a single line comment ***/**,
	2	**/*** this comment begins here and ends here ***/**,
	3	if a(v10) **/***and p(v20) ***/** then v20/ fi,

@	**incluye archivo de formato**
Soporte:	CISIS
Sintaxis:	**@**<filename>
Definición:	Inserta en el formato corriente un formato que ha sido almacenado en un archivo externo.
Notas:	El <filename> puede incluir el dispositivo y la ruta del directorio en donde está alojado el formato. La sintáxis de los comandos contenidos en el archivo, es evaluada cuando el formato corriente es ejecutado. Es necesario encerrar entre comas el llamado a este comando (,@<filename>,).
Ejemplos:	1 **@**test.pft,v20, 2 s(**@**c:\temp\test.pft,v3), 3 if v1='L' then **@**large.pft, fi,

A

a(field selector)	**verifica ausencia de campo**
Soporte:	Standard
Tipo de función:	Boolean
Sintaxis:	**a(**<field selector>**)**
Definición:	Regresa el valor TRUE (verdadero) si el campo está ausente, y FALSE (false) en caso contrario.
Notas:	Pueden usarse todos los componentes del selector de campo, excepto el comando de sangría
Ejemplos:	1 if **a(**v12**)** then v13 else v12, fi, 2 if **a(**v20^b**)** and p(v30) then v40/, fi,
Vea también:	**p** function **v** command

actualiza campo	
Vea:	**proc** function

B

break	**salto u salida condicional**
Soporte:	CISIS
Sintaxis:	**break**
Definición:	Interrumpe la ejecución del formato de un grupo repetible. Cuando está fuera de un grupo repetible, sale del formato que se está ejecutando.
Notas:	La ejecución continuará con los comandos posteriores al grupo repetible. Cuando se utiliza dentro de una **función ref**, la ejecución del formato continúa después de la función.
Ejemplos:	1 (if iocc > 10 then '10+ occurrences'/,**break** else v5^n|-|,v5^s,/, fi,),
Vea también:	**(**format**)** command

búsqueda de llave	
Vea:	**l** function

búsqueda en cadena de caracteres	
Vea:	**instr** function

C

c	**columna**
Soporte:	Standard
Sintaxis:	**c**<int>
Definición:	Desplazarse a una columna específica en la línea actual o en la línea siguiente.
Ejemplos:	1 'Name: ',c10,v1^n/,
	2 if p(v1^s) then c10,v1^s/, fi,
Vea también:	**X** command

cancela línea en blanco	
Vea:	**%** command

cat(file)	**muestra contenido de un archivo**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**cat(**<format>)
Definición:	Muestra el contenido de un archivo cuyo nombre es generado por el <format>.

cat(file)	**muestra contenido de un archivo**
Ejemplos:	1 mfn,cat('myfile.html'),
	2 cat('current document'/, ,if v10='c' then 'firstdoc.txt' else 'default.doc' fi),
	3 cat(v101),
Vea también:	**s** function

comentario	
Vea:	**/*string*/** command

comprueba la ausencia de un campo	
Vea:	**n** command

concatena cadenas de caracteres	
Vea:	**s** function

continue	**salto condicional repetible**
Soporte:	CISIS
Sintaxis:	continue
Definición:	Ejecuta la ocurrencia siguiente de un grupo repetible, si existe al menos un campo con tal ocurrencia.
Notas:	
Ejemplos:	1 (if iocc = 1 then continue else v10/ fi),
	2 (f(iocc,1,0),'=',v70,continue/),
Vea también:	**(**format**)** command

control de flujo condicional	
Vea:	**if ... then ... else ... fi** command

convierte de caractere a numérico	
Vea:	**val** function

convierte de numérico a caractere	
Vea:	**f** function

crea variable en el ambiente	
Vea:	**putenv** function

D

d	**verifica presencia del campo**
Soporte:	Standard
Sintaxis:	**d**<field tag><subfield>
Definición:	Muestra el literal condicional prefijo si el campo o subcampo asociado contiene datos. Se usa asociado a un literal condicional.
Notas:	El selector de campo virtual no retorna un valor. Bugs conocidos: Cuando en un grupo repetible, el <subfield> es evaluado sólo para la primer ocurrencia del campo.
Ejemplos:	1 "this text outputs if data field 10 exists"**d**10, 2 "Name: "v20(5,5)/, ,"Name: "n20,v21(5,5)/,
Vea también:	**"string"** command **n** command **v** command

date **date(keyword)**	**fecha actual**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**date** **date**(<keyword>)
Definición:	Muestra la fecha actual del sistema. Usado sin parámetros, devuelve: **aaaammdd hhmmss d nnn** dónde: aaaa = año mm = mes dd = día hh = hora mm = minuto ss = segundo d = día de la semana (0-6) nnn = cantidad de días transcurridos desde el 1ro de Enero.
Componentes:	keywords **DATETIME** and **DATEONLY**
Notas:	**DATETIME** muestra la fecha del sistema en formato europeo y la hora actual (dd/mm/aa hh:mm:ss), mientras que **DATEONLY** muestra lo mismo pero sin la hora.
Ejemplos:	1 'Today is ',**date**, 2 'Current date: ',**date(DATEONLY)**/, ,'Current time: ',mid(**date(DATETIME)**,10,8)/,

datex	**genera una fecha**
Soporte:	CISIS
Tipo de función:	string
Sintaxis:	datex (<fmt>)

Definición:	Genera una fecha en la forma DATE, equivalente a los segundos desde el 1 enero 1970 00:00:00 generado por <fmt>
Notas:	Se aplican los límites de SECONDS( )
Ejemplo:	mx null "pft= datex(1147780749) /"
Vea también:	**date** command **seconds** function

define el ancho de línea

Vea:	**lw** function

E

e0 .. e9	**define variables numéricas**
Soporte:	CISIS
Tipo de función:	Numérica
Sintaxis:	e<n>:=<expresión numérica>
Definición:	CISIS define 10 variables numéricas e0 .. e9. Las variables se inicializan con valor 0 cada vez que se ejecuta el formato.
Notas:	Puede cambiarse el valor de la variable durante la ejecución del formato. Una variable numérica puede usarse en cualquier parte que se requiera un valor, por ejemplo como operando de una expresión del tipo `if e1+10<25 then ... fi` Al igual que cualquier valor numérico, una variable numérica no puede desplegarse directamente, sino que debe convertirse primero usando la función f
Ejemplos:	e1:=val(v10 + 5)
Vea también:	**s0 ... s9** command **f** function

ejecución condicional en bloque

Vea:	**select ... case ... elsecase ... endsel** command

extrae cadena de caracteres a derecha

Vea:	**right** function

extrae cadena de caracteres a izquierda

Vea:	**left** function

F

f(num expr, length, decimals)	**convierte de numérico a caractere**
Soporte:	Standard
Tipo de función:	String
Sintaxis:	**f(**<format>,<expr-1>,<expr-2>**)**
Definición:	Convierte un valor numérico en cadena de caracteres. <format> es la expresión numérica a ser convertida. <expr-1> y <expr-2> son opcionales y determinan la longitud mínima de la salida y la cantidad de decimales respectivamente.
Notas:	Si el <formato> no es una expresión numérica válida, se reporta un error. Si presente <expr-2> también debe estar <expr-1> o se producirá un error de sintaxis. Si sólo se indica la <expr-1>, el resultado se muestra en notación científica exponencial. Si el número de caracteres requeridos para representar al <formato> es mayor que la <expr-1>, se agregan posiciones automáticamente. Si la <expr-1> está ausente, se asume una extensión de 16 caracteres.
Ejemplos:	1 **f(**val(v1),2**)**, 2 **f(**((3+5)/2)+1,4,2**)**, 3 **f(**v2**)**,
Vea también:	**val** function

fecha actual	
Vea:	**date** function

G

getenv(expression)	**lee variable del ambiente**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**getenv(**<format>**)**
Definición:	Regresa el valor de una variable de ambiente.
Notas:	Si el <formato> no genera el nombre de una variable de ambiente válido, no regresa ningún valor.
Ejemplos:	1 'Current path: ',**getenv(**'PATH'**)**, 2 (v1|=|,**getenv(**v1**)**/),
Vea también:	**putenv** function

grupos repetibles	
Vea:	**(**format**)** command

I

if ... then ... else ... fi	**control de flujo condicional**
Soporte:	Standard
Sintaxis:	**if** <bool expr> **then** <format-1> [**else** <format-2>] **fi**
Definición:	Ejecuta un bloque de lenguaje de formateo (<formato-1>) si la evaluación de <bool expr> retorna TRUE (verdadero). Ejecuta otro bloque de lenguaje de formateo (<formato-2>) usando la cláusula **else** que se ejecuta cuando la evaluación de <bool expr> retorna FALSE (falso).
Notas:	La cláusula **then** precede al primer bloque del formato. **else** es opcional y, si está presente, debe estar seguida de un bloque de lenguaje de formateo. La cláusula **fi** siempre termina el comando. Y, si está ausente, se reporta un error de sintáxis. El comando **if ... fi** puede ocupar varias líneas, en cuyo caso es recomendable usar sangría.
Ejemplos:	1 **,if** instr(v5,'ab')>0 **then** ,v5/, **fi,** 2 **,if** p(v10) **then** ,|Title: |v3, **else** ,|Alternate title: |v4, **,fi,**

incluye archivo de formato	
Vea:	**@** command

índice de la ocurrencia	
Vea:	**iocc** function

inserta espacios	
Vea:	**x** command

instr(string1,string2)	**búsqueda en cadena de caracteres**
Soporte:	CISIS
Tipo de función:	Numeric
Sintaxis:	**instr(**<format-1>,<format-2>**)**
Definición:	Regresa un número especificando la posición inicial del string generado por <format-2>, en el string generado por <format-1>. Si el string buscado (<format-2>) no se encuentra en <format-1> la función retorna cero.
Notas:	Tanto <format-1> como <format-1> deben generar cadenas de caracteres (strings), en caso contrario se producirá un error de sintaxis. El uso de **la función s** puede ayudar en los casos donde se requiere un string complejo como parámetro.

instr(string1,string2)	**búsqueda en cadena de caracteres**
Ejemplos:	1 if **instr(**v5,'ab'**)**>0 then v5/, fi,
	2 if **instr(**s(|'|v1|'|)**,**v5**)**>0 then v1, fi,
	3 left(v18,**instr(**v18,'.'**)**-1),

iocc	**índice de la ocurrencia**
Soporte:	CISIS
Tipo de función:	Numeric
Sintaxis:	**iocc**
Definición:	Regresa el número de orden (índice) de la ocurrencia en proceso (comenzando en 1), o cero en caso contrario.
Ejemplos:	1 ("Author: "v1/, ,if **iocc** > 3 then 'et all',break, fi),
	2 (f(**iocc**,3,0),|.|v10/),
Vea también:	**nocc** function

L

l(key) **l([inverted file],key)**	**búsqueda de llave**
Soporte:	Standard/CISIS
Tipo de función:	Numeric
Sintaxis:	**l(**<format key>**)** **l(**[<format ifname>]<format key>**)** **l->**ifname(<format key>) (compatible con Winisis de Unesco)
Definición:	Regresa el **MFN** del primer posting (si lo hubiera) usando la clave generada por el formato <format key> para buscar en el archivo invertido actual. Puede también buscar en otro archivo invertido cuyo nombre se indica mediante el formato <format ifname>.
Notas:	Las claves se convierten a mayúsculas antes de buscar la expresión. El modo de visualización por defecto es mpl. Si se especifica un modo diferente en la FST, debe tenerse en cuenta en <format key> ya que este genera la clave. Si no se encuantra la clave, la función retorna cero. El parámetro <format ifname> debe generar un string con un nombre de archivo invertido válido, de otra manera ocurrirá un error de sintaxis. Esta función también se usa frecuentemente en conjunción con la función ref para permitir la visualización de campos de otro registro.
Ejemplos:	1 if **l(**v15**)**<> 0 then |Term: |v15, fi,
	2 ref(**l(['books']**v1,'-',v2**)**,v10/),
	3 **ref->books(l->books**(v1,'-',v2),v10/),
Vea también:	**ref** function

left(string,length)	**extrae cadena de caracteres a izquierda**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**left(**<format-1>,<format-2>**)**
Definición:	Retorna un nuevo string, que contiene los n primeros caracteres del string original <format-1>, comenzando desde la izquierda, la cantidad n de caracteres es determinada por <format-2>.
Notas:	Si el valor obtenido del string generado por <format-2> es mayor que el tamaño del string generado por <format-1>, la función retorna el string <format-1>. Si <format-2> es cero o un número negativo, retorna un string NULO.
Ejemplos:	1 if **left(**v1^n,2**)**='Ma' then v1^n/, fi, 2 **left(**v1,instr(v1,'.')-1**)**,
Vea también:	**right** function **mid** function

literal condicional	
Vea:	**"string"** command

literal condicional para grupo repetible	
Vea:	**|string|** command

literal incondicional	
Vea:	**'string'** command

lw(number)	**define el ancho de línea**
Soporte:	CISIS
Tipo de función:	Numeric
Sintaxis:	**lw(**<int>**)**
Definición:	Establece que el ancho de la línea de salida es de <int> caracteres.
Notas:	El ancho por defecto de la línea de salida es de 76 caracteres.
Ejemplos:	1 if size(v10) > 76 then **lw(**254**)**, fi, 2 **lw(**70**)**,v20/,**lw(**10**)**,v30/,

M

mdl, mdu, mhl, mhu, mpl, mpu	**modo de formatación**
Soporte:	Standard
Sintaxis:	**m<mode><conv>**
Definición:	Establece un nuevo modo de visualización para la salida actual.
Notas:	El modo por defecto es mpl. <mode> representa el modo que se desea establecer. <conv> especifica si se establece la conversión a mayúsculas. MODE puede aparecer varias veces en un formato y su efecto sobre el formateo estará activo hasta que se establezca un nuevo modo. Para **<mode>** puede especificarse de las siguientes maneras: **p** = proof: los campos se visualizan tal como se almacenaron en los registros. **h** = heading: los caracteres de control y los delimitadores de campo son ignorados, excepto los delimitadores de subcampo, que son reemplazados por signos de puntuación. **d** = dato: similar al modo encabezamiento, solo que agrega un punto al final del campo, seguido de dos espacios. En **<conv>** se pueden establecer las siguientes opciones: **u :** convierte los datos a mayúscula (u de upper case) **l** : deja los datos en minúscula (l de lower case), en realidad como estaban.
Ejemplos:	1 **mpl**,"First author: "v10[1]/, 2 **mpu**,"Second author: "v10[2]/, 3 **mdl**,"Third author: "v10[3]/,

mfn mfn(length)	**número del registro**
Soporte:	Standard
Tipo de función:	String or numeric
Sintaxis:	**mfn** **mfn(**<int>**)**
Definición:	Regresa el MFN (Master File Number) de un registro.
Notas:	Un valor entero puede pasarse como parámetro para establecer la longitud del string que retornará la función MFN. **mfn** regresa un valor de tipo númerico ó cadena dependiendo de los requerimientos del formato.
Ejemplos:	1 'Record: ',**mfn(**3**)**/, 2 if **mfn** > 2 then **mfn**/, fi, 3 ref(**mfn**-1,v2/),
Vea también:	**ref** function **l** function

mid(string, start, length)	**regresa parte de una cadena de caracteres**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**mid(**<format-1>,<format-2>,<format-3>**)**
Definición:	Regresa un nuevo string, conteniendo un número específico de caracteres del string original (<format-1>). <format-3> espesifica la cantidad de caracteres a leer de <format-1> y <format-2> la posición a partir de la cual se extraen los caracteres de <format-1>.
Notas:	Si <format-2> es mayor que el tamaño de <format-1>, la función regresa un string NULO. Si <format-2> es cero o un número negativo, el valor por defecto es 1.
Ejemplos:	1 **mid(**v2,2,80**)**, 2 **mid(**v1,instr(v1,'key'),size(v1)**)/**,
Vea también:	**right** function **left** function

modo de formatación	
Vea:	**mdl, mdu, mhl, mhu, mpl, mpu** command

mstname	**nombre del archivo maestro**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**mstname**
Definición:	Regresa el nombre del archivo maestro actual/corriente.
Ejemplos:	1 'Current database: ',**mstname**/, 2 ref(['names']l(['names']'X39BJ'), ,'Database now is ',**mstname**/),

muestra contenido de un archivo	
Vea:	**cat** function

N

n	**comprueba la ausencia de un campo**
Soporte:	Standard
Sintaxis:	**n**<field tag>
Definición:	Comprueba la ausencia de un campo. Se utiliza en conjunción con literales condicionales.
Notas:	Como selector de campo virtual, no regresa ningun valor.
Ejemplos:	1 "this text outputs if data field 10 is absent"**n**10, 2 "Author:"v10/, ,"Author: "**n**10,v20/,

n	**comprueba la ausencia de un campo**
Vea también:	**"string"** command **d** command **v** command

newline(string)	**establece / restablece caractere de nueva línea**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**newline(**<format>**)**
Definición:	Establece y/o restablece el par CR/LF por defecto con el/los caracter(es) generados por <format>.
Notas:	<format>, puede también contener una secuencia de escape reservada como: \r - es un salto de párrafo \n - es un salto de línea Los saltos de párrafo y línea \ subsecuentes serán reemplazados automaticamente por el string generado por <format> hasta que un nuevo llamado a la función **newline** establezca un nuevo par de strings (o caracteres) para salto de línea y párrafo.
Ejemplos:	1 **newline**(if v151='unix' then '\n' else '\r\n' fi, 2 **newline**(v301), 3 **newline**(' '),
Vea también:	**/** command **#** command

nocc(field)	**número de ocurrencias**
Soporte:	CISIS
Tipo de función:	Numeric
Sintaxis:	**nocc(**<field selector>**)**
Definición:	Regresa el número de ocurrencias de un campo o subcampo. El campo o subcampo se indica mediante <field selector>.
Notas:	Esta función sólo puede recibir como parámetro un <field selector> con campos o subcampos. Todos los demás componentes de la función <field selector>, produciran un error de sintaxis si son utilizados.
Ejemplos:	1 if **nocc(**v3**)**> 10 then 'Too many occurrences.'/, fi, 2 'There are ',f(**nocc(**v20**)**,2,0),' authors.'/,
Vea también:	**iocc** function **v** command

nombre del archivo maestro	
Vea:	**mstname** function

npost(key) **npost([inverted file],key)**	**postings de llave**
Soporte:	CISIS
Tipo de función:	Numeric
Sintaxis:	**npost(**<format key>**)** **npost(**[<format>],<format key>**)**
Definición:	Regresa todos los postings de una clave (la cual es generada por <format key>) en un archivo invertido. Si <format> es indicado, este debe generar un string conteniendo el nombre del archivo invertido a ser utilizado. <format key> genera la clave a buscar en el archivo invertido.
Ejemplos:	1 if **npost(**v1**)**> 1 then 'duplicate key ',v1,' found'/, fi, 2 'There are ',f(**npost(**v20**)**,3,0),'keys for ',v20,'. '/,
Vea también:	**l** function

nueva línea condicional	
Vea:	**/** command

nueva línea incondicional	
Vea:	**#** command

número de ocurrencias	
Vea:	**nocc** function

número del registro	
Vea:	**mfn** function

P

p(field selector)	**verifica presencia de campo**
Soporte:	Standard
Tipo de función:	Boolean
Sintaxis:	**p(**<field selector>**)**
Definición:	Retorna TRUE (verdadero) si el campo asociado está presente, retorna FALSE (falso) en caso contrario.
Notas:	Pueden utilizaese todos los componentes de la función field selector (**v**) excepto sangría.
Ejemplos:	1 if **p(**v12**)** then v12 else v13, fi, 2 if **p(**v50^a**)** and **p(**v50^b**)** then v50^a/,v50^b/, fi,
Vea también:	**a** function **v** command

postings de una llave

Vea: **npost** function

proc(field update format) — actualiza campo

Soporte: CISIS

Tipo de función: String

Sintaxis: **proc(**<fldupd format>**)**

Definición: Añade o reemplaza campos en el registro actual. <fldupd format> es un formato que genera los comandos de actualización que especifican a la función las tareas a realizar.

Notas: Una especificación de actualización de campos es un string (cadena de caracteres) compuesto por los comandos **d** (borrar), **a** (agregar) y **h** (agregar) y las modificaciones que estos especifiquen. Las modificación se aplicará al registro corriente. Todos los comandos **d** (borrar) deben preceder a los comandos agregar (**a** y **h**).
Especificación de los comandos:
d* - borra todos los campos del registro
d<field tag> - borra todas las ocurrencias del campo <field tag>
d<field tag>**/**<occ> - borra la ocurrencia <occ> del campo <field tag>
a<field tag>**#<**string>**#** - agrega la cadena <string> como una nueva ocurrencia del campo <field tag>
h<field tag> <n> <string> - agrega la cadena <string>, de <n> bytes de longitud, como una nueva ocurrencia del campo <field tag>
El delimitador **#** puede ser cualquier caracter no numérico.
Debe haber un espacio entre <field tag>, <n> y <string>, parámetros del comando **h**.

Ejemplos:

1 **proc(**'d70',|a10#|v70|#|**)**,

2 **proc(**if v24*0.4 = 'Tech' then 'd*', fi**)**,

putenv(expression) — crea variable en el ambiente

Soporte: CISIS

Tipo de función: String

Sintaxis: **putenv(**<format>**)**

Definición: Establece una variable de ambiente a el nivel del sistema operativo con su valor correspondiente.

Notas: La variable está disponible solo dentro del alcance del proceso actual.

Ejemplos:

1 **putenv(**'TEST=test'**)**,getenv('TEST'),

2 set CIPAR=somefile
set
mx null "pft=**putenv(**'CIPAR=another'**)**,getenv('CIPAR')/"
set

Vea también: **getenv** function

R

ravr(string) — valor promedio de expresión

Soporte:	Standard
Tipo de función:	Numeric
Sintaxis:	**ravr(**<format>**)**
Definición:	Regresa el valor promedio de un formato dado. <format> debe generar una expresión alfanumérica.
Notas:	Puede usarse para computar el promedio de valores numéricos en campos repetibles.
Ejemplos:	1 f(**ravr(**s(v8,x1,v1)**)**,3,0), 2 f(**ravr(**v1,x1,v2**)**,5,2), 3 f(**ravr(**'8;15;16.73'**)**,3,2), 4 if **ravr(**v20|;|**)**>=5 then 'pass'/ else 'fail'/, fi,
Vea también:	**rmin** function **rmax** function **rsum** function

ref(mfn, format)
ref([master file]mfn, format) — ejecuta formato en el registro seleccionado

Soporte:	Standard/CISIS
Tipo de función:	String
Sintaxis:	**ref(**<expr>,<format>**)** **ref(**[<format dbname>]<expr>,<format>**)** **ref->dbname**(<expr>,<format>) (compatible con Winisis de Unesco)
Definición:	Ejecuta <format> en el registro seleccionado mediante <expr>. Si se indica <format dbname> se puede referenciar a otra (o la misma) base de datos.
Notas:	<expr> puede ser cualquier formato que regrese el MFN de un registro. La función **l** puede usarse para ejecutar una búsqueda y retornar el MFN del primer registro encontrado.
Ejemplos:	1 **ref(**l(v3),v1/,v2/,v3/**)**, 2 if **ref(**['account']l(['user']v2),v4**)**='active' then |Name: |v10/, fi, 3 (if p(v99) then **ref(**[v99]1,v30/**)**, fi), 4 **ref->books**(**l->books**(v1,'-',v2),v10/),
Vea también:	**l** function

regresa parte de una cadena de caracteres

Vea:	**mid** function

replace(string1, string2, string3) — cambio de cadenas de caracteres

Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**replace(**<format-1>,<format-2>, <format3**)**
Definición:	Devuelve una nueva cadena de caracteres (string), reemplazando <format-2> con <format-3> en <format-1>
Notas:	Si <format-2> es una cadena de caracteres (string) nula o no se encuentra en <format-1>, la función devuelve la cadena <format-1>. Si <format-3> es nula, la cadena <format-2> será excluida de <format-1>. **Replace** es una función sensible para ambas cadenas de caracteres: la cadena a buscar (<format-2>) y la cadena de reemplazo (<format-3>).
Ejemplos:	1 **replace(**'Mary And John','And','and'**)/**, 2 if **replace(**v1^a,'01x','01X'**)**= '894501X' then v1^n/, fi, 3 replace(s(v304,v333),',',', ')/, 4 **replace(**s(if v415='spanish' then v299 else 'none' fi),v1,v759**)/**,

right(string, length) — extrae cadena de caracteres a derecha

Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**right(**<format-1>,<format-2>**)**
Definición:	Retorna un nuevo string, que contiene los n últimos caracteres del string original <format-1>, comenzando desde la derecha; la cantidad n de caracteres es determinada por <format-2>.
Notas:	Si el valor de <format-2> es mayor que la longitud de <format-1>, la función devuelve la cadena de caracteres <format-1>. Si <format-2> es igual a cero o contiene un número negativo, no devuelve nada.
Ejemplos:	1 if **right(**v1^n,1**)** = 'r' then v1^n/, fi, 2 right(v65,4)/,

rmax(string) — valor máximo de una expresión

Soporte:	Standard
Tipo de función:	Numeric
Sintaxis:	**rmax(**<format>**)**
Definición:	Devuelve el valor máximo de un formato dado. <format> debe generar una cadena de caracteres (string).
Notas:	Puede ser utilizado para calcular el máximo entre los valores numéricos de un campo repetible.
Ejemplos:	1 f(**rmax(**'72,54,2'**)**,2,0), 2 f(**rmax(**v1,x1,v4,x1,(v8|,|)**)**,5,2), 3 if **rmax(**v40|;|**)**>val(v41) then 'Limit of ',v41,'exceeded.'/, fi,

rmax(string) — valor máximo de una expresión

Vea también:
rmin function
ravr function
rsum function

rmin(string) — valor mínimo de una expresión

Soporte: Standard

Tipo de función: Numeric

Sintaxis: **rmin(**<format>**)**

Definición: Devuelve el valor mínimo del formato especificado. <format> debe generar una cadena de caracteres (string).

Notas: De forma similar a la **función rmax**, rmin puede calcular el mínimo de valores numéricos en un campo repetible.

Ejemplos:

1 f(**rmin(**'10;2;5;4;-2'**)**,2,0),

2 f(**rmin(**v1,x1,v2,x1,'44'**)**,4,2),

3 if **rmin(**v80||,v90| |,v100| |**)** < 1990 then 'Wrong decade.'/, fi,

Vea también:
rmax function
ravr function
rsum function

rsum(string) — suma de un formato

Soporte: Standard

Tipo de función: Numeric

Sintaxis: **rsum(**<format>**)**

Definición: Devuelve la suma de un formato dado. <format> debe generar una cadena de caracteres (string).

Notas: En forma similar a las **funciones rmax** y **rmin**, rsum calcula la suma de valores numéricos en un campo repetible.

Ejemplos:

1 f(**rsum(**'102,45,-37'**)**,2,0),

2 f(**rsum(**v1,x1,v3,x1,f(val(v8)+2)**)**,4,2),

3 if **rsum(**v20^d**)**>1000 then 'Aborted.'/ else 'OK'/, fi,

Vea también:
rmax function
ravr function
rmin function

S

s0 ... s9	**define variable**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	s<n>:=(<fmt>)
Definición:	CISIS define 10 variables string s0 .. s9. Las variables se inicializan como strings nulas cada vez que se ejecuta el formato.
Notas:	Los paréntesis alrededor del formato son obligatorios. Una variable string puede usarse tanto como un operando como un comando de formato.
Ejemplos:	s1:=('CDS/ISIS') s3:=(v10)
Vea también:	**e0 .. e9** command **f** function

s(expression)	**concatena cadenas de caracteres**
Soporte:	Standard
Tipo de función:	String
Sintaxis:	**s(**<format>**)**[command component]
Definición:	Devuelve la concatenación de cadenas de caracteres (string) generadas por <format>.
Componentes:	extraction
extraction:	Extracts partial content of the resulting string. <offset int> is the first position to start extraction, while <length int> determines how many characters will be extracted. If <length int> is omitted or is greater than the resulting string, the default is the end of the resulting string.
Notas:	Puede ser utilizadas por funciones que requieran una cadena de caracteres como parámetro.
Ejemplos:	1 if **s(**v1,v2,v3**)**:'ABCDE' then **s(**v1,v2,v3**)***0.50, fi, 2 if **s(**|*|v5|*|**)**:**s(**'*E*'**)**then 'English'/, fi,
Vea también:	**v** command

seconds	**calcula el número de segundos**
Soporte:	CISIS
Tipo de función:	numeric
Sintaxis:	seconds(<fmt>)

seconds	**calcula el número de segundos**
Definición:	Función numérica para calcular el número de segundos desde 1 enero 1970 00:00:00 hasta la fecha generada por <fmt> con alguno de los siguientes formatos ‘aaaammdd’ ‘aaaammdd hh’ ‘aaaammdd hhmm’ ‘aaaammdd hhmmss’
Notas:	La fecha generada por <fmt> debe estar en el rango ‘19700102 000000’ a ‘20380118 031407’ Un día contiene 24 x 60 x 60 = 86400 segundos
Ejemplos:	mx null “pft=date/f(seconds(date) – seconds(20010305 172915’),1,0)” va a desplegar 20010305 172916 1 63 1.. 20010305 172916 1 63 2.. 20010305 172916 1 63 3..
Vea también:	**date** command **datex** function

select ... case ... elsecase ... endsel	**ejecución condicional en bloque**
Soporte:	CISIS
Sintaxis:	**select** <format expr> **case** <option-1>: <format-1> **case** <option-2>: <format-2> **case** <option-n>: <format-n> [**elsecase** <format-0>] **endsel**
Definición:	Se evalua <format expr> y se compara el resultado con cada opción **case** (<option-1>, <option-2>...<option-n>). Si una opción es coincide con <format expr>, se ejecuta el bloque de instrucciones asociado (<format-1>, <format-2>...<format-n>); si ninguna opcion es igual a <format expr> se ejecuta la cláusula **elsecase** (<format-0>), si fué definida.
Notas:	<format expr> debe generar una cadena de caracteres (string) o un valor numérico. Si <format expr> genera una cadena, todos los valores de las opciones de las clausulas **case** deben se de tipo cadena de caracteres, de lo contrario. Si <format expr> genera un valor numérico, los valores de la opción también deben ser numéricos.

select ... case ... elsecase ... endsel — ejecución condicional en bloque

Ejemplos:

1 **select** s(v5)
case '1'**:** ,f(val(v5)/2,2,2)/,
case '2'**:** ,v5/,
case '3'**:** ,v6,'-',v1/,
elsecase ,|Error in field v5 = |v5/,
endsel,

2 **select** nocc(v7)
case 0: 'absent'/,
case 1: 'one occurrence'/,
case 2: 'two occurrences'/,
elsecase 'more than 2 occurrences'/,
endsel,

Vea también: **if ... then ... else ... fi** command

selector de campo

Vea: **v** command

size(string) — tamaño de la cadena de caracteres

Soporte: CISIS

Tipo de función: Numeric

Sintaxis: **size(**<format>**)**

Definición: Devuelve el tamaño de una cadena de caracteres (string).

Notas: <format> debe devolver una cadena de caracteres (string) de lo contrario se producirá error de sintaxis.

Ejemplos:

1 if **size(**v10**)**> 76 then lw(254), fi,

2 f(**size(**v10,v20),1,0**)**,

suma de un formato

Vea: **rsum** function

system(expression) — ejecuta comando del sistema operativo

Soporte: CISIS

Tipo de función: String

Sintaxis: **system(**<format>**)**

Definición: Ejecuta el argumento producido por <format> como un comando del sistema operativo.

Notas: <format> debe generar una cadena de caracteres (string) que contenga el comando a ser ejecutado. La eventual salida generada por este comando será direccionada a la salida estandar del sistema.

Ejemplos:

1 **system(**'dir'**)**,

2 if p(v2) then **system(**'type ',v2**)**, fi,

T

tamaño de una cadena de caracteres	
Vea:	**size** function

tipo del contenido del formato	
Vea:	**type** function

type(string)	**tipo del contenido del formato**
Soporte:	CISIS
Tipo de función:	String
Sintaxis:	**type(<format>)**
Definición:	Devuelve el tipo de una cadena de caracteres de la siguiente manera: A - si la cadena contiene solamente caracteres alfabéticos (conforme a una tabla de caracteres alfabéticos por defecto, como ISISAC.TAB)o espacios N - si la cadena contiene únicamente caracteres numéricos (0-9) X - para cualquier otro caso.
Notas:	<format> debe generar una cadena de caracteres (string) o un mensaje de que ha ocurrido un error de sintaxis.
Ejemplos:	1 if **type(**v1**)**='N' then f(val(v1),3,2)/ else v1/, fi, 2 if s(**type(**v1**)**,**type(**v2**)**,**type(**v3**)**)<>'AAA' then 'Invalid character type detected'/, fi,

V

v	**selector de campo**
Soporte:	Standard
Sintaxis:	**v**<field tag>[command components]
Definición:	Contenido de los campos de salida de datos. El contenido puede ser seleccionado, restringido, extraido o sangrado mediante componentes del mismo comando (ver más abajo). **v** significa campo de longitud variable.
Componentes:	subfield, occurence, extraction and indent
syntactic order:	**^**<subfield id> **[**<index>**[..**<upper index>**]]** *****<offset int>**.**<length int> **(**<first line int>**,**<next line int>**)**
subfield:	Restricts the output to the contents of a subfield. If data field exists but subfield is not present, no output is generated.

v	**selector de campo**
occurrence:	Narrows the output to one or a range of occurrences of a repeatable field. <index> and <upper index> refer to the first (or unique) and last occurrences, respectively. If the specified <index> is greater than the actual number of occurrences, no output is generated. The same occurs if data field is not repeatable and <index> is set to a number equal or greater than 2. However, if <index> is set to 1 and it is used in a non-repeatable field, content is normally output. This component must be used outside a repeatable group; otherwise, <upper index> is ignored. If double dot (**..**) is used and <upper index> is missing **LAST** is assumed. The **LAST** keyword is set with the value of total occurrences of a data field.
extraction:	Extracts partial content of a data field, subfield or occurrence. <offset int> is the first position to start extraction, while <length int> determines how many characters will be extracted. If <length int> is omitted or is greater than field length, the default is the end of data field.
indent:	Aligns the output of data field, subfield, occurrence or extracted content, according to <first line int> (alignment for the first line) and <next line int> (alignment for successive lines). Both values are numeric constants. If current line position differs from zero, indentantion is disabled.
Notas:	El funcionamiento del comando **v** depende de los componentes utilizados. No se generará salida alguna cuando el campo de datos esté ausente o cuando el componente ejecuta una restricción o una extracción que esté fuera de los límites.
Ejemplos:	1 **v**2/,**v**3^a| - |,**v**1/, 2 **v**1^n*0.3, 3 (|; |+**v**3^s)/, 4 **v**20**[**4**]**, 5 **v**10**[**2..7**]**/, 6 **v**5**[**3**..]**/,/* equals to ,**v**5**[**3**..LAST]**, */ 7 **v**1[**LAST**]*****2**.**7/, 8 **v**1**(**5,5**)**/, 9 |Title: |**v**1^n**(**5,5**)**/,
Vea también:	**"string"** [conditional literal] **d** [dummy field selector] **n** [not present] **|string|** [repeatable conditional literal] **(**format**)** [repeatable group]

val(string)	**convierte de caractere a numérico**
Soporte:	Standard
Tipo de función:	Numeric
Sintaxis:	**val(**<format>**)**
Definición:	Devuelve el valor numérico del argumento generado por <format>.

val(string)	**convierte de caractere a numérico**
Notas:	Si <format> produce solamente caracteres no numéricos, la función devuelve cero. Si encuentra más de un valor numérico, sólo el primero de ellos es devuelto.
Ejemplos:	1 if **val(**v2**)**>5 then 'Error'/ else 'OK'/, fi, 2 f(**val(**v2**)**/3,4,2),

valor máximo de una expresión	
Vea:	**rmax** function

valor mínimo de una expressión	
Vea:	**rmin** function

valor promedio de expresión	
Vea:	**ravr** function

verifica presencia del campo	
Vea:	**d** command

W

while	**control condicional de ejecución**
Soporte:	CISIS
Sintaxis:	**while** <condition> **(**<fmt>**)** Condition: es una expresión booleana fmt: es un formato CISIS que se ejecuta mientras la expresión booleana es True
Definición:	El comando WHILE permite ejecutar repetidamente un formato (loop)
Notas:	Si el valor inicial de la condición es False, entonces el formato no se ejecutará. Para que la ejecución termine debe incluirse en fmt algún comando necesario para que convierta la condición en False. De otro modo podrá generarse un ciclo (loop) infinito y el sistema no responderá al usuario.

Ejemplos:	e1:=1,
	e2:=nocc(v16),
	while e1<=e2
	(f(e1,1,0),'. ', v16[e1]/
	e1:=e1+1,
	),
	El ejemplo despliega cada ocurrencia del campo v16 (autor) precedido por el número de la ocurrencia.
	`primer autor`
	`segundo autor`
	`tercer autor`

Vea también:	**if ... then ... else ... fi** command **select ... case ... elsecase ... endsel** command

X

x	**inserta espacios**
Soporte:	Standard
Sintaxis:	**x**<int>
Definición:	Inserta una cantidad <int> de espacios.
Notas:	Si <int> es más grande que el espacio disponible en la línea en curso, pasa a la línea siguiente.
Ejemplos:	1 'Name: ',**x5**,v1^n/, 2 (v1,**x3**,v2,**x8**,v3/),
Vea también:	**c** command

Citas bibliográficas

1. UNESCO. *Mini-micro CDS/ISIS:* Reference manual (version 2.3). Organized by Giampaolo Del Bigio. Paris: United Nations Educational, Scientific and Cultural Organization, 1989. 286 p. ISBN 92-3-102-605-5.

2. BUXTON, Andrew, HOPKINSON, Alan. *The CDS/ISIS for Windows Handbook* [online]. Paris: United Nations Educational, Scientific and Cultural Organization, 2001 [cited 30 August 2006]. 164 p. Available from internet: <http://bvsmodelo.bvs.br/download/winisis/winisis-handbook-en.pdf>.

3. SUTER, Tito. “Prehistoria” e historia del MicroISIS [online]. In: *Manual para instructores de Winisis*. Buenos Aires: Centro Atómico Constituyentes (CAC), Comisión Nacional de Energía Atómica (CNEA), 1999 [citado el 30 Agosto 2006]. p. 21-26. Disponible en internet: <http://www.cnea.gov.ar/cac/ci/isis/isidams.htm>.

Glosario

- **Archivo**. En computación, un conjunto de datos que se puede grabar en algún dispositivo de almacenamiento. Los archivos de datos son creados por aplicaciones, como por ejemplo un procesador de textos.

- **Archivo invertido**. Conjunto de seis archivos físicos, cinco de los cuales contienen los términos de búsqueda del diccionario (organizados como un árbol B*) y el sexto contiene la lista de apuntadores asociadas a cada término. A fin de optimizar el almacenamiento en disco, se mantienen dos árboles B* por separado: uno para los términos de hasta 10 caracteres (almacenados en los archivos *.N01* y *.L01*) y otro para los términos de más de 10 caracteres (almacenados en los archivos *.N02* y *.L02*). El archivo *.CNT* contiene campos de control para ambos árboles B*). En cada archivo del árbol B* el archivo *.N0x* contiene los nodos del árbol y el archivo *.L0x* contiene las hojas. Los registros de las hojas apuntan al lugar donde se encuentran los apuntadores que contienen la información para localizar los registros (postings) en la base de datos. Este archivo se identifica con la extensión *.IFP*.

- **Backup**. Procedimiento en el que uno o más archivos y/o directorios son duplicados para otro dispositivo de almacenamiento (cinta o disco), para producir una copia de seguridad, que puede restaurarse en el caso de que algún dato sea borrado accidentalmemnte o si ocurrió daño físico de los datos originales.

- **Base de datos**. Colección de datos estructurados para que sea posible acceder a ellos y manipularlos fácilmente. Está formada por unidades denominadas registros, cuyos diversos atributos son representados por campos y subcampos. Por ejemplo, en un archivo "catastro de clientes", cada cliente representa un registro, que posee varios campos, como "NOMBRE", "CÓDIGO DEL CLIENTE", "TELÉFONO" etc.

- **Bases de datos bibliográfica**. Versión electrónica de un catálogo o índice bibliográfico.

- **Campo**. Elemento de un registro que permite almacenar información específica. Ver *Base de datos*.

- **CDS/ISIS – MicroISIS**. Software desarrollado y mantenido por la UNESCO para el tratamiento de datos bibliográficos.

- **Clave**. Expresión que identifica una o más informaciones de determinada clase o tipo y que puede ser usada en la busqueda.

- **Formato de presentación**. Conjunto de comandos que determinan como debe ser la salida de datos de una base de datos ISIS.

- **Formato electrónico**. Cualquier forma de almacenamiento, recuperación y presentación de información pasible de transmisión online o grabación en medios magnéticos u ópticos.

- **Formato ISO (de intercámbio de datos)**. Patrón establecido por la ISO para intercambio de datos entre instituciones, redes y usuarios. Se refiere la norma ISO 2709.

- **Formato LILACS**. Formato de descripción bibliográfica establecido por BIREME, basado en la UNISIST Reference Manual for Machine-readable Bibliographic Descriptions.

- **Indización**. Procedimiento de identificar y describir el contenido de un documento con términos que representan los temas correspondientes a ese documento, con el objetivo de recuperarlo posteriormente.

- **Posting**. Consiste de la dirección de una clave extraída del archivo maestro.

- **Registro**. Conjunto estructurado de datos que permite almacenar determinado asunto. Ver *Base de datos*.

- **Subcampo**. Elemento que contiene la menor parte de información de un campo, cuyo sentido puede no ser claro si no fuera analizado en conjunto con los otros elementos relacionados.

- **UNISIST**. Programa intergubernamental relativo a las cooperaciones en el campo de la información científica y tecnológica.